Dynamic Resource Allocation for Overload Avoidance and Green Cloud Computing

Saima Israil
Computer Science and Engineering
UIT, RGPV, Bhopal, India

Rajeev Pandey, PhD
Assistant Professor,
Depart. of CSE
UIT, RGPV, Bhopal, India

Uday Chourasia
Assistant Professor,
Depart. of CSE
UIT, RGPV, Bhopal, India

ABSTRACT

Cloud Computing is a flourishing technology because of its scalability, flexibility, availability of resources and other features. Resource multiplexing is done through the virtualization technology in cloud computing. Virtualization technology acts as a backbone for provisioning requirements of the cloud based solutions. At present, load balancing is one of the challenging issues in cloud computing environment. This issue arises due to massive consumer demands variety of services as per their dynamically changing requirements. So it becomes liability of cloud service provide to facilitate all the demanded services to the cloud consumers. However, due to the availability of finite resources, it is very challenging for cloud service providers to facilitate all the demanded services efficiently. From the cloud service provider's perspective, cloud resources must be allocated in a fair manner. This paper addresses the existing techniques for resource allocation in cloud computing and proposes the dynamic resource allocation technique to mitigate overloads. It also focuses on energy consumption issue of cloud data centres and devised technique for lower energy consumption in order to achieve green cloud computing.

Keywords

Cloud computing, dynamic resource allocation, overload avoidance, green computing.

1. INTRODUCTION

Cloud computing has become more and more popular with the widely deployment of several cloud infrastructures [1]. The underlying principle of cloud computing is to deliver the required services from shared hardware through virtualization technology. The goal of this computing model is to make a better use of distributed resources, put them together to make higher throughput and to handle large-scale computation problem efficiently and economically. Cloud computing can been broadly categorized into three levels of use model or cloud computing services.

Infrastructure-as-a-service (IaaS): Cloud computing replaces mainly computer hardware. Users of IaaS can manage to support operating systems and applications, but don't desire to buy server, storage and networking hardware and a data centre to house the hardware. Examples of those providers are companies such as Amazon, ENKI, GoGrid[2].

Platform-as-a-service (PaaS): Cloud computing replaces an execution environment for a computer language by providing a system ready to execute the user's software. The user of PaaS is the programmer. Examples of those providers are companies such as Engine Yard or Google [3].

Software-as-a-Service (SaaS): The cloud user interacts directly with the Cloud software provided by CSP and often pays for usages only in place of computer time. Examples of those providers are NetSuite, SalesForce.com, Google Apps [4].

2. OVERVIEW OF THE CLOUD RESOURCE ALLOCATION PROBLEM

In cloud computing environments, efficient resource provisioning and management is a challenging task for cloud service provider. Dynamically changing needs of the cloud users and the need to satisfy heterogeneous resource requirements further exacerbate the resource allocation management concern. In such a dynamic environments where cloud users can connect or disconnect at any time, there shall be provision by the cloud service provider to be able to make accurate decisions for scaling up or down its data-centers resources [5].

The resource allocation technique responds to find the resource allocation solution that satisfies a specific goal of the cloud service provider. While managing the resource numerous utility criteria shall be considered like delay of virtual resources setup, migration of existing processes, the resource utilization, energy consumption, overload avoidance, minimize resource wastage, ensuring service level agreement etc. So it become essential to allocate the resources appropriately but the static allocations have some constraints. Dynamic resource allocation can overcome these constraints [6]. Using the virtualization techniques, virtual machines can migrate to physical machines effectively [7].In order to resolve the resource allocation issue and to fulfil the could service provider and the end-users requirements, an efficient and dynamic resource allocation strategy becomes mandatory.

In view of essential characteristics of cloud and specific requirements for dynamic resource management, in this paper, we provide an overview of the recent research advancement in cloud computing dynamic resource management and propose the efficient methodology to overcome overload problem while minimizing the physical machines used to cater the user demands. Simulation was done based on proposed methodology and satisfactory result was observed for overload avoidance and shaving of energy consumption was also envisaged.

3. RELATED WORK

According to our knowledge, we have not noticed any comprehensive journal article on IaaS cloud Dynamic resource allocation approaches. However, a number of related research papers and reviews that referred to IaaS cloud resource allocation have been published. In this section, we describe relative mechanisms and the methods which are implemented earlier and also the advantages and disadvantages of each method are described briefly.

Ying Song *et al.* [8] has proposed A two-tiered on-demand resource allocation mechanism, including the local and global resource allocation, based on a two-level control model. A well designed on demand resource allocation algorithm may minimize the waste of resources as well as guarantee the quality of the hosted applications. The local on-demand resource allocation on each server optimizes the resource allocation to VMs within a server taking the allocation threshold into account, while the global on-demand resource allocation optimizes the resource allocation among applications at the macro level by adjusting the allocation threshold of each local resource allocation.

Christopher Clark *et al.* [9] Live OS migration is an extremely powerful tool for cluster administrators, allowing separation of hardware and software considerations, and consolidating clustered hardware into a single coherent management domain. If a physical machine needs to be removed from service an administrator may migrate OS instances including the applications that they are running to alternative machine(s), freeing the original machine for maintenance. Similarly, OS instances may be rearranged across machines in a cluster to relieve load on congested hosts. In the situations, the combination of virtualization and migration significantly improves manageability. Live migration refers to the process of making running virtual machines or applications between different physical machines without disconnecting the client or application. Memory, storage and network connectivity of the virtual machines are transferred from the original host machine to the destination.

Marvin McNett *et al.* [10] in his paper reveal that Usher provides a simple abstraction of a logical cluster of virtual machines, or virtual cluster. Usher users can create any number of virtual clusters of arbitrary size, while Usher multiplexes individual virtual machines on available physical machine hardware. Two modules are there, first modules enable Usher to interact with broader site infrastructure, such as authentication, storage, and host address and naming services. Second, pluggable modules enable system administrators to express site-specific policies for the placement, scheduling, and use of VMs. As a result, Usher allows administrators to decide how to configure their virtual machine environments and determine the appropriate management policies.

Xiaoyun Zhu *et al.* [11] Auto Control a resource control system that automatically adapts to dynamic changes in a shared virtualized infrastructure to achieve application SLOs. Auto Control is a combination of an online model estimator and a novel multi-input, multi-output resource controller. The model estimator captures the complex relationship between application performance and resource allocation, while the MIMO controller allocates the right amount of resources to achieve application SLOs. Virtualization is causing a disruptive change in enterprise data centres and giving rise to a new paradigm: shared virtualized infrastructure. In this new paradigm, multiple enterprise applications share dynamically allocated resources. These applications are also consolidated to reduce infrastructure and operating costs while simultaneously increasing resource utilization. As a result, data centre administrators are faced with growing challenges to meet service level objectives in the presence of dynamic resource sharing and unpredictable interactions across many applications.

Gong Chen *et al.* [12] Load skewing algorithms that allow significant amount of energy saving without sacrificing user experiences, i.e. maintaining very small number of SIDs. Understanding how power is consumed by connection servers provides insights on energy saving strategies. Connection servers are CPU, network, and memory intensive servers. There is almost no disk IO in normal operation, except occasional log writing. Since memory is typically preallocated to prevent run-time performance hit, the main contributor to the power consumption variations of a server is the CPU utilization .if pack connections and login requests to a portion of servers, and keep the rest of servers hibernating, here it can achieve significant power savings. However the consolidation of login requests results in high utilization of those servers, which may downgrade performance and user experiences. Hence, it is important to understand the user experience model before address the power saving schemes for large-scale Internet service.

T. R. Gopalkrishnan Nair *et al.* [13] presented a model, named as Ruled Based Resource Allocation (RBRAM) which deals with the efficient resource utilization in M-P-S (Memory-Processor-Storage) Matrix Model. Authors say that resource allocation rate should be greater than resource request rate. Major components of the system are: cloud priority manager, cloud resource allocation, virtualization system manager and end result collection. To analyse the performance of the cloud system authors considered the Cloud Efficiency Factor. However, authors also identified other parameters of Cloud System for future work.

Justin Y. Shi *et al.* [14] explored a simple quantitative Timing Model method for cloud resource planning. For the same they considered the estimated resource usage times in steady state. Authors had calculated Speed up for Parallel Resource Planning based on Parallel Matrix Multiplication. To investigate multiple important dimensions of a program's scalability, authors proposed quantitative application dependent instrumentation method instead of qualitative performance models. Authors had mainly focused on application inter dependencies for cost effective processing.

Chu-Fu Wang *et al.* [15] in "A Prediction Based Energy Conserving Resources Allocation Scheme for Cloud Computing", has develop an Energy Conserving Resource Allocation Scheme with Prediction (ECRASP) for cloud computing systems. The prediction mechanism can predict the trend of arriving jobs (dense or sparse) in the near future and their related features, so as with help the system to make adequate decisions. Simulation results show that our proposed ECRASP method performs well compared to conventional resource allocation algorithms in the energy conserving comparisons.

Stefan Spitz *et al.* [16] authored paper in which he present approaches which improve current trust models according to the problems mentioned. Thereby, the degree of automation in the trust evaluation process increased. Finally, in combination with an adjusted trust level workflow, the presented approach allows an optimal resource allocation for grid or cloud computing service providers in combination with a trust model, a service provider can evaluate a resources' performance based on a set of trust and QoS requirements. As a result, trust relations can be established between a service provider and the associated resources. This allows the service provider to assign resources which are not only capable of processing a given task but also will most likely perform well.

4. SYSTEM OVERVIEW

The data flow diagram of the proposed system is presented in Figure 1. Each physical machine (PM) runs the Xen hypervisor (VMM) [7]. It is assumed that all PMs Share backend storage. The multiplexing of VMs to PMs is managed using the Usher framework [8]. Each node runs an Usher which collects the usage statistics of resources for each VM on that node. The statistics collected at each PM are forwarded to the Usher central controller (Usher CTRL) where our VM scheduler runs.

Modules description

VM Scheduler: VM Scheduler run and invoked periodically receives the resource demand history of VMs (virtual machines), the capacity and the load history of PMs (physical machines), and the current layout of VMs on PMs. Then it can forward the request to predictor.

Predictor: The predictor predicts the future resource demands of VMs and the future load of PMs based on past statistics. The load of a PM is computed by aggregating the resource usage of its VMs. Xen can change the CPU allocation among the VMs by adjusting their weights in its CPU scheduler.

Hot spot Solver: The hot spot solver in VM Scheduler detects if the resource utilization of any PM is above the hot threshold (i.e., a hot spot). If so, some VMs running on them will be migrated away to reduce their load. Then it can give the request to cold spot solver.

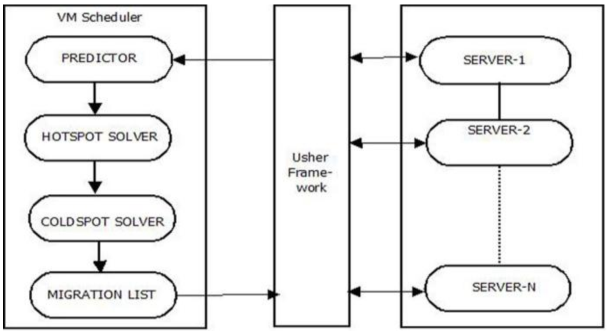


Fig 1 Dynamic Resource allocator modules

Cold spot Solver: The cold spot solver checks if the average utilization of actively used PMs (APMs) is below the green computing threshold. If so, some of those PMs could potentially be turned off to save energy. It identifies the set of PMs whose utilization is below the cold threshold (i.e., cold spots) and then attempts to migrate away all their VMs then it forward request to migration list

Migration List: When migration list can receive the request from cold spot solver and it can compiles list of VMs and migration list can passes it response to the Usher CTRL (user controller) for execution

4.1 Resource usage balancing

The concept of skewness has been used to quantify the unevenness in the utilization of multiple resources on a server. Let n be the number of resources we consider and r_i be the utilization of the i^{th} resource. The resource skewness of a server p is defined as

$$Skewness(Sp) = \sqrt{\sum_{i=1}^{n} \left(\frac{r_i}{\bar{r}} - 1\right)^2}$$

Where, $\bar{r}$ is the average utilization of all resources for server p. In practice, not all types of resources are performance critical and hence we only need to consider bottleneck resources in the above calculation. By minimizing the skewness factor, different types of workloads can be pooled nicely and improve the overall utilization of server resources.

4.2 Hot and Cold Spots

The algorithm executes periodically to evaluate the resource allocation status based on the predicted future resource demands of VMs. The server can be defined as a hot spot if the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VMs running on it should be migrated away. The temperature of a hot spot T_p as the square sum of its resource utilization beyond the hot threshold:

$$temperature(Tp) = \sum_{r \in R} (r - r_t)^2 \text{ if, } r > r_t, \text{ else } T_p = 0$$

Where R is the set of overloaded resources in server p and r_t is the hot threshold for resource r.

The temperature of a hot spot reflects its degree of overload. If a server is not a hot spot, its temperature is zero. Server is defined as a cold spot if the utilizations of all its resources are below a cold threshold. This indicates that the server is mostly idle and a potential candidate to turn off to save energy. However, server shall be turned off only when the average resource utilization of all actively used servers (i.e., APMs) in the system is below a green computing threshold. A server is actively used if it has at least one VM running. Otherwise, it is inactive. Finally, the warm threshold can be defined as the a level of resource utilization that is sufficiently high to justify having the server running but not high enough as to risk becoming a hot spot in the face of temporary fluctuation of application resource demands.

4.3 Hot Spot Mitigation

The list of hot spots in the system is sorted out and arranged in descending temperature (i.e., the hottest one first). The goal is to eliminate all hot spots if possible. Otherwise, keep their temperature as low as possible. For each server p, it is to be decided first that which of its VMs should be migrated away. The list of VMs is sorted based on the resulting temperature of the server if that VM is migrated away. Those VM will be selected to migrate away the VM that can reduce the server's temperature the most. In case of ties, that VM will be selected whose removal can reduce the skewness of the server the most. For each VM in the migration list, suitable host server for destination VM has to be selected to accommodate it. The server must not become a hot spot after accepting this VM. Among all such servers, selecting the one whose skewness can be reduced the most by accepting this VM. Note that this reduction can be negative which means the selection of the server whose skewness increases the least. If a destination server is found, the migration of the VM recorded to that server and updates the predicted load of related servers. Otherwise, iteration will be done the next VM in the list and try to find a destination server for it. As long as the destination server can be found for any of its VMs, this run of the algorithm is considered as a success and then move onto the next hot spot. Note that each run of the algorithm migrates away at most one VM from the overloaded server. This does not necessarily eliminate the hot spot, but at least reduces its temperature. If it remains a hot spot in the next decision run, the algorithm will repeat this process. It is possible to design the algorithm so that it can migrate away multiple VMs during each run. But this can add more load on the related servers during a period when they are already overloaded. It has been optimistically decided to use this more conservative approach and leave the system some time to react before initiating additional migrations.

4.4 Green Computing

When the resource utilization of active servers is too low, some of them can be turned off to save energy. This is handled in the green computing algorithm. The challenge here is to reduce the number of active servers during low load without sacrificing performance either now or in the future. It is needed to avoid oscillation in the system.

The green computing algorithm is invoked when the average utilizations of all resources on active servers are below the green computing threshold. The list of cold spots is sorted in the system based on the ascending order of their memory size. Since it is needed to migrate away all its VMs before we can shut down an underutilized server, the memory size of a cold spot is defined as the aggregate memory size of all VMs running on it. Recall that our model assumes all VMs connect to shared back-end storage. Hence, the cost of a VM live migration is determined mostly by its memory footprint. It is tried to eliminate the cold spot with the lowest cost first.

For a cold spot *p*, check if all its VMs can be migrated somewhere else. For each VM on *p*, try to find a destination server to accommodate it. The resource utilizations of the server after accepting the VM must be below the warm threshold. While energy can be saved by consolidating underutilized servers, overdoing it may create hot spots in the future. The warm threshold is designed to prevent that. If multiple servers satisfy the above criterion, preference is given to one that is not a current cold spot. This is because increasing load on a cold spot reduces the likelihood that it can be eliminated. However, cold spot can be accepted a as the destination server if necessary. All things being equal, the destination server will be selected whose skewness can be reduced the most by accepting this VM. If the destination servers for all VMs on a cold spot can be found, the sequences of migrations are recorded and update the predicted load of related servers. Otherwise, any of its VMs will not be migrated. The list of cold spots is also updated because some of them may no longer be cold due to the proposed VM migrations in the above process.

The above consolidation adds extra load onto the related servers. This is not as serious a problem as in the hot spot mitigation case because green computing is initiated only when the load in the system is low. Nevertheless, extra load due to server consolidation is desired to be bounded. The number of cold spots are restricted that can be eliminated in each run of the algorithm to be no more than a certain percentage of active servers in the system. This is called the consolidation limit. Note that we eliminate cold spots in the system only when the average load of all active servers (APMs) is below the green computing threshold. Otherwise, those cold spots are left as it is there as potential destination machines for future offloading. This is consistent with philosophy that green computing should be conducted conservatively.

4.5 Consolidated Movements

The movements generated in each step above are not executed until all steps have finished. The list of movements is then consolidated so that each VM is moved at most once to its final destination. For example, hot spot mitigation may dictate a VM to move from PM A to PM B, while green computing dictates it to move from PM B to PM C. In the actual execution, the VM is moved from A to C directly.

4.6 Live Migration for Overloaded Machines

Now let us consider the live migration scenario for overloaded machines [17, 18]. The general steps for it are as follows

a. *Monitor Resource Usage*

A monitoring policy should be made to monitor the resource usage of the each server on the cloud.

b. *Check If The Resource Usage Is Beyond The Hot Threshold*

Then a checking should be maintained which will let know the provision system that the upper threshold limit i.e hot threshold of the host is crossed and the host is over utilized.

c. *Select The VM*

Then VM which is causing overload should be discovered and selected for migration.

d. *Select The Destination*

Select suitable host for this selected VM which maintain minimum skewness with resources needs for migrating VM.

e. *Move : VM To Destination Host*

VM is moved to destination host for its placement on the new host.

4.7 Live Migration of Underutilized Machines

Now, consider the live migration scenario for overloaded machines. The general steps for it are as follows.

a. *Monitor Resource Usage*

A monitoring policy should be made to monitor the resource usage of the each server on the cloud.

b. *Check If The Resource Usage < Cold Threshold*

Then a checking should be maintained which will let know the provision system that the lower threshold limit of the host is crossed and the host is underutilized.

c. *Select All VM*

Then select all the VMs from the underutilized host.

d. *Select The Destination*

Once all VMs are selected, there is a need to find out which is the capable host of this VM with resources needed by migrating VM.

e. *Move All VM To Destination Host*

After the host is selected for the VM, VM is moved to destination host for its placement on the new host.

f. *Switch Of The Source Host*

After all the VMs are migrated from the source host, the host should be switched off in order to save energy.

5. SIMULATION

The performance evaluation is based on the trace driven simulation. This ensures the fidelity of our simulation results. Traces are per-minute server resource utilization, such as CPU rate, memory usage, and network traffic statistics [6].

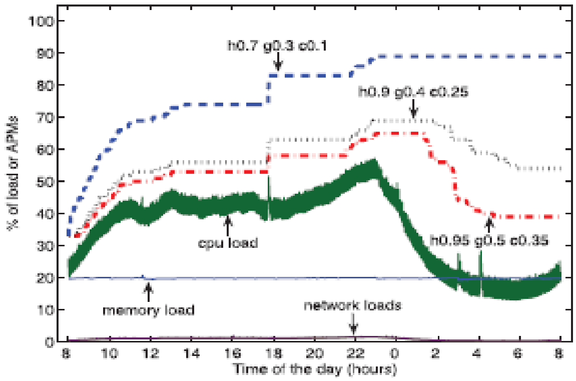


Fig 2 CPU Utilization, Memory & Network Utilization [6]

Simulation was run using the NetBeans, Number of PM and VM was fed to configuration window invoked with threshold parameter setting for CPU resource threshold, Memory resource threshold, Cold Threshold and Warm threshold. Hot Spot threshold is chosen as 80 % of either resource utilization.

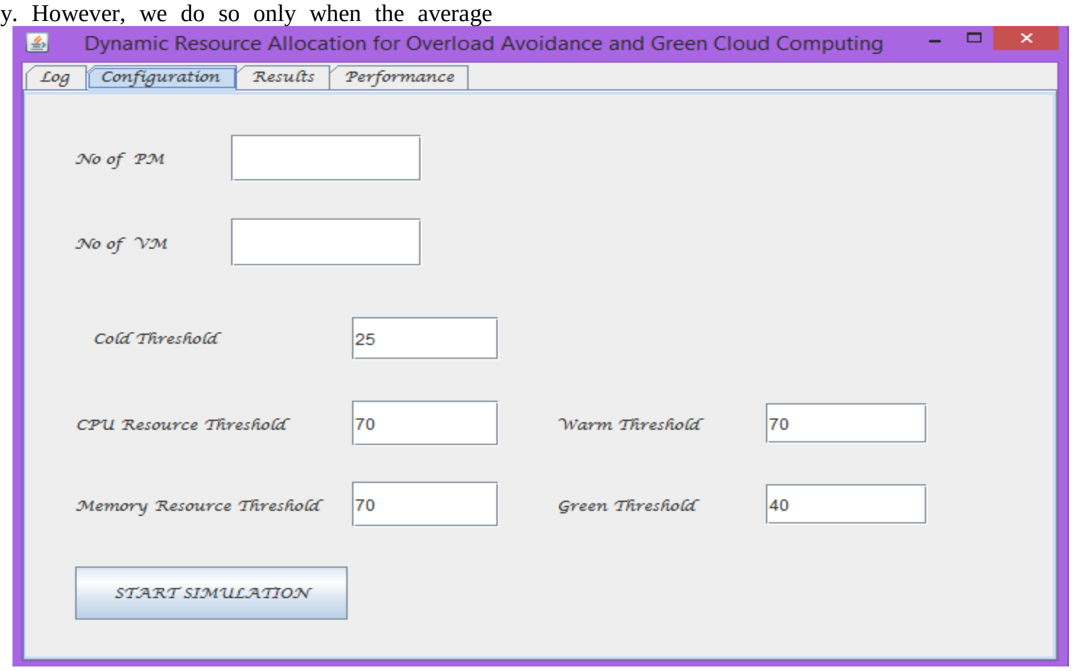
5.1 Parameters in our Simulation

A server is defined as a cold spot if the utilizations of all its resources are below a cold threshold. This indicates that the server is mostly idle and a potential candidate to turn off to save energy. However, we do so only when the average resource utilization of all actively used servers (i.e., APMs) in the system is below a green computing threshold. A server is actively used if it has at least one VM running. Otherwise, it is inactive. Finally, we define the warm threshold to be a level of resource utilization that is sufficiently high to justify having the server running but not as high as to risk becoming a hot spot in the face of temporary fluctuation of application resource demands.

Different types of resources can have different thresholds. For example, we can define the hot thresholds for CPU and memory resources to be 80 and 70 percent, respectively. Thus a server is a hot spot if either its CPU usage is above 80 percent or its memory usage is above 70 percent.

Table 1: Simulation parameter

Symbol	meaning	value
h	Hot threshold	0.8
w	Warm threshold	0.7
g	Green computing threshold	0.4
c	cold threshold	0.25


Dynamic Resource Allocation for Overload Avoidance and Green Cloud Computing
Log
Configuration
Results
Performance
START SIMULATION

No of PM

No of VM

Cold Threshold: 25

CPU Resource Threshold: 70

Warm Threshold: 70

Memory Resource Threshold: 70

Green Threshold: 40

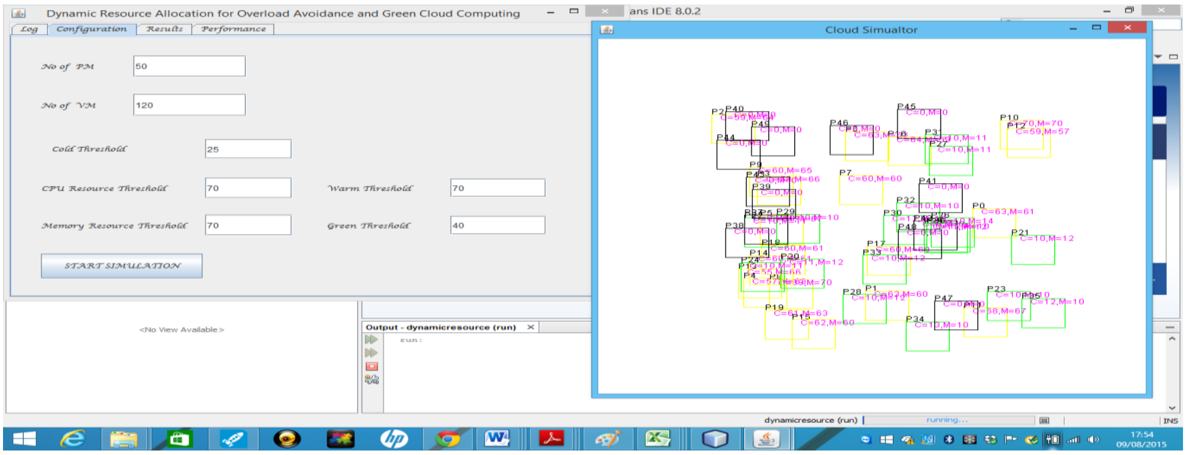
Fig. 3 Start simulation of Dynamic resource allocation window

6. SIMULATION RESULTS

Simulation was initialized with 30 PM and 90 VM and all threshold setting as parameter value in above table 1 for simulation parameter. Hot spot observed during initial stage of simulation shown as Red colour rectangles. Some VMs from PMs having hotspot are migrated utilizing dynamic resource allocation algorithm and VM placement to suitable host was done. Further cloud simulator shows yellow rectangles as the number of active physical machines (APM), Green PMs as physical machine which can further be turned off to safe energy and achieve green computing. Black rectangles show the currently Inactive PM which is turned off as a result of cold spot migration.

Further, several simulation was carried out with varying the number of PM and VM in order to obtain the performance curve for active PMs, Migrations, hot spot and energy consumption in existing system and proposed system for resource allocation in cloud environment.

6.1 Screen Shots



Dynamic Resource Allocation for Overload Avoidance and Green Cloud Computing

Log | Configuration | Results | Performance

No of PM: 50

No of VM: 120

Cold Threshold: 25

CPU Resource Threshold: 70

Warm Threshold: 70

Memory Resource Threshold: 70

Green Threshold: 40

START SIMULATION

<No View Available>

Output - dynamicresource (run)

```
run:
```

Fig 4: Simulation with 50 PM, 120 VM

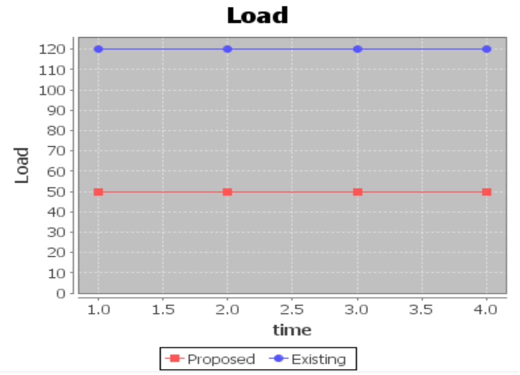


Fig 5: Load on data centre with VM 300, PM120

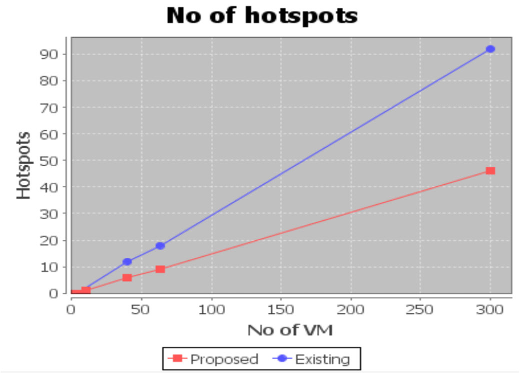


Fig 6: Observed hotspots

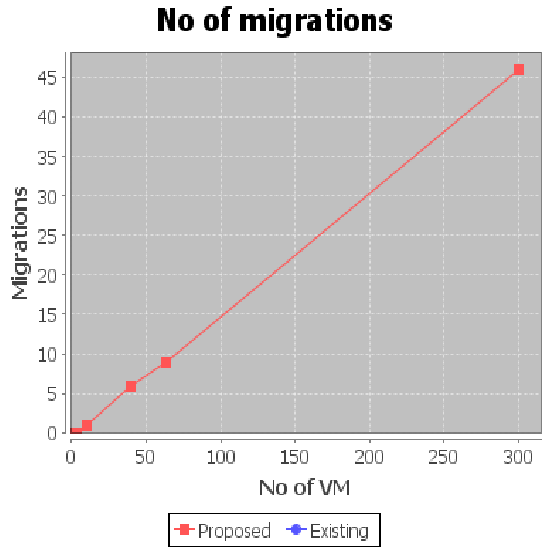


Fig 7: Number of Migrations

6.4 Result Interpretation

Varying the number of physical machines and Virtual machines at data centre performance significant reduction in data centre physical machines been observed.

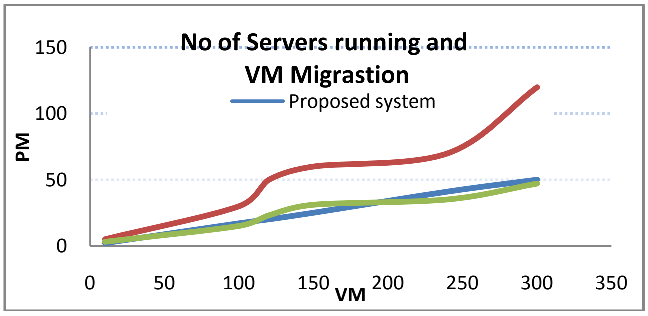


Fig 9: Loaded PM in existing and proposed system

Graph above show the Number of Virtual Machines running on physical machines in cloud data centre for existing and proposed system. Proposed resource allocation strategy has been seen as around 50-60% less servers is utilized for corresponding number of virtual machine as compared to the existing system. The number consolidated migration of VM machines is also shown in above graph as a result of dynamic resource allocation algorithm. Thus the number of server in service has been optimized accordingly with the VM requirements.

It has been observed that while utilizing our algorithm for dynamic resource allocation and live migration the power consumptions by servers has reduced significantly. Around 55% of energy saving can be achieved by minimizing the server in use. Moreover lesser number of server running will have lesser heat generation consequently the cooling system requirement of data centre will be reduce significantly which further adds to saving in energy consumption of data centre to achieve green cloud computing.

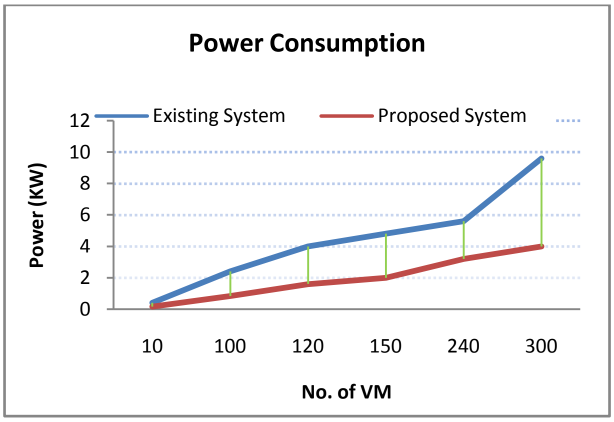


Fig 8: Power consumption by servers

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

Design, implementation, and evaluation of a resource management system for cloud computing services is presented. The system multiplexes virtual to physical resources adaptively based on the changing demand. The skewness metric is used to combine VMs with different resource characteristics appropriately so that the capacities of servers are well utilized. Proposed algorithm achieves both overload avoidance and green computing for systems with multi-resource constraints. By switching of idle machines power can be saved at cloud data centers. Live migration helps in saving power by migration of the virtual machines causing overload and underutilization of the host. Simulation shows around 55 % of Energy saving utilizing our resource allocation approach.

7.2 Future Work

There are a number of open research challenges that need to be addressed in order to further advance the area. Hot threshold and cold threshold is static value chosen which closely resembling the loading pattern of CPU, Threshold value affect the VM placement on sever, so dynamic threshold calculation can be implemented in future work for effective resource utilization of servers. Further dimension of green cloud computing can be studied and implemented during design stages for efficient green cloud computing hardware aspect and efficient cooling system for data centres.

8. REFERENCES

[1]. Rimal, B.P., Choi, E., Lumb, I., 2009, A Taxonomy and Survey of Cloud Computing Systems, Proceeding of the Fifth International Joint Conference on INC, IMS and IDC, pp. 44 – 51.

[2]. http://aws.amazon.com/ec2/ ; http://www.enki.co/ ; http://www.gogrid.com/ available at:Amazone ec2.

[3]. http://www.engineyard.com/products/cloud ; http://www.google.com/apps/intl/en/business/cloud.html, available at:Engine yard.

[4]. http://www.netsuite.com/portal/home.shtml;http://www.salesforce.com; https://developers.google.com/, available at:Netsuite.

[5]. Saima Israil, Dr. Rajeev Pandey, Prof. Uday Chaurasia "Survey on Dynamic resource allocation techniques for Overload avoidance and green cloud computing", SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE) –Volume 2 Issue 3 March 2015.

[6]. Zhen Xiao, Senior member, IEEE, weijia song and Qi chen "Dynamic Resource allocation using Virtual Machines For Cloud Computing Environment," IEEE Transaction on parallel and distributed systems, vol.24, No.6 june 2013.

[7]. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, Xen and the Art of Virtualization," Proc. ACM Sy mp. Operating Systems Princip les Oct. 2003.

[8]. Ying Song, Yuzhong Sun, Member, IEEE, and Weisong Shi, Senior Member, IEEE "A Two-TieredOn-Demand Resource Allocation Mechanism for VMBased Data Centers", IEEE t ransactions on services computing, vol. 6, no. 1, january-march 2013.

[9]. Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, Andrew Warfield "Live Migration of Virtual Machines",University of Cambridge Computer Laboratory 15 JJ Thomson Avenue, Cambridge, UK.

[10]. Marvin McNett, Diwaker Gupta, Amin Vahdat, and Geoffrey M. Voelker "Usher: An Extensible Framework For Managing Clusters Of Virtual Machines", Proceedings of Large Installation System Administration Conference 2007 pp. 167-181.

[11]. PradeepPadala, Kai-Yuan Hou Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, SharadSinghal, Arif Me rchant "Automated Control of Multiple Virtualized Resources", The University of Michigan, Hewlett Packard Laboratories.

[12]. Gong Chen, Wenbo He, Jie Liu, SumanNath, Leonidas Rigas, Lin Xiao, Feng Zhao "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services",Dept. of Computer Science, University of Illinois, Urbana-Champaign, IL 61801.

[13]. T.R. Gopalkrishnan Nair, Vaidehi M, "Efficient Resource Arbitation And Allocation Stratargies In Cloud Computing Through Virtualization" in Proceedings of IEEE CCIS2011, 978-1-61284-204-2/11.

[14]. Justin Y. Shi, Moussa Taifi and Abdallah Khreishah, "Resource Planning for Parallel Processing in the Cloud" in IEEE International Conference on High Performance Computing and Communications, 978-0-

7659-4538-7/11, Nov. 2011.

[15]. Wang Chu-Fu, Wen-Yi, Hung, and Yang Chen-Shun, "A Prediction Based Energy Conserving ResourcesAllocation Scheme for Cloud Computing"2014 IEEE International Conference on Granular Computing (GrC), 978-1-4799-5464-3/14 , 2014 IEEE, pp.321-324.

[16]. Stefan S, Patrick B , York T"Trust-based Resource Allocation and Evaluation of Workflowsin Distributed Computing Environments",2010 2nd International Conference on Software Technology and Engineering(ICSTE) 978-1-4244-8666-3/10, 2010 IEEE, pp.IV-372-76.

[17]. C Clark, K Fraser, S Hand, J G Hanseny, E July,C Limpach, I Pratt, A Wareld ,"Live Migration of Virtual Machines" NSDI'05 Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation ,2005.

[18]. E Arzuaga, D R Kaeli, "Quantifying load imbalance on virtualized enterprise servers." In WOSP/SIPEW '10: Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering, ACM, 2010.